Free Questions for Rev-Con-201

Shared by Roman on 16-04-2026

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

A solution architect notices that a complex product bundle uses multiple nested Constraint Modeling Language (CML) rules to enforce constraints during product configuration. Users report long load times when adding options to the bundle.

What should the architect do to improve configuration performance and ensure quotes remain technically and commercially viable?

Options:

A- Refactor and simplify CML constraints to reduce runtime complexity.
B- Disable constraint rules during configuration and validate selections after deployment.
C- Replace CML entirely with Apex triggers to enforce constraints at quote submission.

Answer:

A

Explanation:

Comprehensive and Detailed From Exact Extract:

Revenue Cloud documentation recommends optimizing and simplifying constraint rules when performance issues arise. Key guidance includes:

Reducing the number of nested and overlapping CML rules.

Consolidating rules when possible.

Avoiding unnecessary complexity that increases runtime evaluation time.

Disabling rules (B) breaks guardrails during configuration and goes against best practices; users would be able to configure invalid combinations.

Replacing CML with Apex triggers (C) removes real-time configurator guidance and is not recommended; constraints should be enforced in the configurator, not only at submission.

Product Configurator and CML Documentation -- Performance and Best Practices

Revenue Lifecycle Management Implementation Guide -- Constraint Rules Optimization

==========

Question 2

Question Type: MultipleChoice

A product bundle has defined a constraint model that is currently in use and has been actively sold for the last few months. A new product will be launched next month and will be sold as part of the same bundle. The product designer updated the bundle structure under Product Catalog Management to add the new product.

What must the product designer do to ensure that the child product is added to the constraint model within the product bundle?

Options:

A- Use the Visual Builder to add the child product to the constraint model, then import the associations for the type from Product Catalog Management.
B- Once a bundle is updated in Product Catalog Management, create a constraint model, then import the associations from Product Catalog Management.
C- Create a new type for the child product in the Constraint Modeling Language (CML) Editor, then import the associations for the type from Product Catalog Management.

Answer:

C

Explanation:

(150--250 words)

In Salesforce Revenue Cloud, Constraint Models define configuration logic through Constraint Modeling Language (CML). When a new child product is added to an existing bundle, the system does not automatically include it in the associated constraint model. To bring the new product into scope, the designer must:

Create a new type in the CML Editor that corresponds to the newly added product.

Import the product associations from Product Catalog Management to ensure the constraint model references the correct product hierarchy and rules.

This approach integrates the new component into the model while retaining existing logic and rule structure. Option A incorrectly assumes the Visual Builder can auto-import relationships for new products, and option B would create a new constraint model rather than updating the existing one.

Exact Extract from Salesforce CPQ Implementation Guide:

''When new products are added to an existing bundle, define a new type for the product in the Constraint Model and import the associations from Product Catalog Management to ensure inclusion in configuration rules.''

Salesforce CPQ Implementation Guide --- Constraint Model Maintenance and CML Editor Usage

Salesforce Revenue Cloud Catalog Management Guide --- Updating Bundles with New Components

Salesforce Solution Architect Handbook --- Constraint Model Versioning and Governance

Question 3

Question Type: MultipleChoice

A Revenue Cloud Consultant needs to display a list of products to be shown in the browse phase of a guided selling journey. Which Product Catalog Management business API should the consultant use to retrieve a list of products that belong to a specific catalog?

Options:

A- Products List (GET)
B- Product Related Records List (POST)
C- Products List (POST)

Answer:

C

Explanation:

The Products List (POST) API is the correct choice for retrieving a list of products belonging to a specific catalog during the browse phase of guided selling in Revenue Cloud. This API endpoint is specifically designed as a composite API for Product Discovery and provides comprehensive filtering capabilities.

According to the Revenue Cloud Developer Guide, the Products List (POST) resource is located at /connect/cpq/products and accepts POST requests with a JSON body. This API allows consultants to specify multiple parameters including catalogId, categoryId, priceBookId, productClassificationId, and various filtering criteria. The POST method is preferred over GET because it can handle complex request bodies with multiple filter criteria, user context information, and qualification/pricing procedures.

The API supports essential Product Discovery features such as enableQualification and enablePricing flags, which are critical during the browse phase. It can also include contextDefinition and contextMapping parameters to ensure proper data flow during guided selling. The Products List (POST) returns a paginated list of products with complete details including pricing information, qualification status, and catalog associations.

Option A (Products List GET) does not exist as a standard Product Catalog Management business API. Option B (Product Related Records List POST) is used for retrieving related records like ProductRampSegment or ProductUsageGrant, not for product lists. The Products List (POST) API is explicitly documented in the Product Discovery Business APIs section of the Revenue Cloud Developer Guide for browsing and discovering products during the sales transaction process.

Question 4

Question Type: MultipleChoice

A Revenue Cloud Consultant needs to update the flow used to browse the catalog. The consultant made the changes and activated the flow but is not seeing the changes during testing.

Which step did the consultant miss?

Options:

A- Adding the Flow Name in the Pricing Settings Setup.
B- Adding the Flow Name in the Product Discovery Setup.
C- Adding the Flow Name in the Revenue Settings Setup.

Answer:

B

Explanation:

Exact Extracts from Salesforce Revenue Cloud (CPQ and Subscription Management Implementation Guides):

''Product Discovery Setup defines the flows used for product selection and catalog browsing in Revenue Cloud.''

''When a custom or updated Product Discovery Flow is created, it must be added to Product Discovery Setup to be invoked during catalog navigation.''

''Activating a flow alone does not make it available for catalog usage unless it is registered in Product Discovery Setup.''

Step-by-Step Reasoning:

Scenario: Consultant modifies and activates a catalog browsing flow but does not see changes in UI.

Requirement: The new or edited flow must be registered within the Product Discovery Setup page to replace the default one.

Why B is Correct:

Product Discovery Setup tells Revenue Cloud which flow to call when browsing products.

Pricing Settings and Revenue Settings relate to pricing and billing, not discovery flows.

Salesforce CPQ Implementation Guide --- Product Discovery and Catalog Setup

Salesforce Subscription Management Implementation Guide --- Flow Configuration for Product Discovery

Question 5

Question Type: MultipleChoice

A Revenue Cloud project requires that a contract agreement dynamically hide or show clauses based on Account-specific fields.

Which token should the implementation consultant use on the document template to show the appropriate clauses at runtime?

Options:

A- Variable tokens
B- Conditional evaluation tokens
C- Repeating content tokens

Answer:

B

Explanation:

Comprehensive and Detailed From Exact Extract:

In Revenue Cloud CLM / DocGen:

Conditional evaluation tokens are used to conditionally include or exclude sections or clauses based on data (for example, Account fields, contract fields).

They evaluate conditions at generation time and determine whether a clause block is rendered.

Variable tokens (A) insert values but do not control visibility. Repeating content tokens (C) are used for lists or repeated items, not conditional visibility.

Document Template & Clause Authoring Guide -- Conditional Tokens for Dynamic Clauses

==========

Question 6

Question Type: MultipleChoice

An administrator is configuring a subscription product in Revenue Cloud for a customer whose order starts on January 1. The finance team wants to collect payment before the service begins so that each invoice is sent on the 15th of the prior month.

Which Billing Type should the Billing Treatment Item have to meet this requirement?

Options:

A- Arrear
B- Advance
C- None

Answer:

B

Explanation:

Exact Extracts from Salesforce Billing Implementation Guide:

''Billing Type determines when a customer is invoiced and charged for a subscription or service. You can set Billing Type to Advance or Arrear.''

''Advance billing collects payment before the service period begins. The invoice is generated prior

to the start date of the service period.''

''Arrear billing collects payment after the service has been delivered.''

Step-by-Step Reasoning:

Requirement: The finance team wants invoices to be sent before the service begins (on the 15th of the prior month).

Advance Billing Type: Ensures that invoicing occurs prior to the service period start, aligning perfectly with the ''collect before service'' requirement.

Arrear Billing Type: Would bill after the service delivery, which does not meet the requirement.

None: Would result in no defined billing schedule, leaving the invoicing behavior undefined.

Therefore, Advance Billing Type is correct.

Salesforce Billing Implementation Guide --- Billing Treatments and Billing Type

Salesforce Subscription Management Implementation Guide --- Invoice Timing and Billing Configuration

Question 7

Question Type: MultipleChoice

A Revenue Cloud Consultant is helping a customer service operations manager determine which product lines have the most order fallout. Which steps should the consultant follow to enable the relevant dashboards?

Options:

A- Within Revenue Settings, enable Revenue Management Intelligence and Install the Dynamic Revenue Orchestrator Analytics App.
B- Within Revenue Management Intelligence Setup, complete Data Cloud Configurations and install the Dynamic Revenue Orchestrator Analytics App.
C- Within Tableau Unified Analytics, create a recipe based on the FulfillmentFalloutRule object and design a dashboard with product line filter.

Answer:

B

Explanation:

To enable dashboards that identify product lines with the most order fallout, the consultant must follow the Revenue Management Intelligence setup process documented in Salesforce Help. The correct approach involves accessing Revenue Management Intelligence Setup (not general Revenue Settings), completing necessary Data Cloud Configurations, and installing the Dynamic Revenue Orchestrator Analytics App.

Revenue Management Intelligence provides a comprehensive suite of Tableau Einstein dashboards specifically designed to assess revenue strategies and achieve cost-effective results. The Dynamic Revenue Orchestrator Analytics App includes specialized dashboards for order fulfillment analysis, including order fallout metrics by product line. These dashboards leverage data collected during the order fulfillment orchestration process.

The setup process requires completing Data Cloud Configurations first. Data Cloud serves as the unified data platform that ingests data from Revenue Cloud transactions, order fulfillment activities, and decomposition events. The configuration ensures that fulfillment data, including fallout metrics, flows properly into the analytics layer. Once Data Cloud is configured, installing the Dynamic Revenue Orchestrator Analytics App provisions the pre-built dashboards that analyze fulfillment performance and identify fallout patterns.

Option A references Revenue Settings, which is used for general Revenue Cloud enablement but not specifically for analytics setup. Option C mentions Tableau Unified Analytics and FulfillmentFalloutRule object, but this approach requires custom development and is not the out-of-the-box solution. The documented out-of-the-box approach through Revenue Management Intelligence Setup with Data Cloud Configurations and the Dynamic Revenue Orchestrator Analytics App is the correct path for accessing pre-built fallout analysis dashboards.

Question 8

Question Type: MultipleChoice

Universal Containers (UC) sells its products and services to other businesses, and provides an automatic discount to businesses that buy in bulk. UC is now expanding its selling channels and plans to sell directly to end users. A key requirement is to ensure that bulk discounts are only applicable to businesses and not individual buyers.

How should the Revenue Cloud Consultant solve this requirement?

Options:

A- By using Quote Transaction Type and Volume-Based Pricing

B- By using Sales Transaction Type and Volume-Based Pricing

C- By using Order Transaction Type and Volume-Based Pricing

Answer:

B

Explanation:

''Transaction Types define how Revenue Cloud interprets pricing, eligibility, and discount logic for a specific transaction (Sales, Amendment, Renewal, Cancellation, etc.).''

''Volume-Based Pricing applies tiered or bulk discounts based on quantities but can be conditioned by Transaction Type.''

''For initial sales, use the Sales Transaction Type. For renewals or amendments, different types apply.''

Step-by-Step Reasoning:

Requirement: Apply bulk discounts only for business-to-business (B2B) sales (initial purchases).

Solution:

Configure Volume-Based Pricing rules.

Condition them on Sales Transaction Type (so they apply only during initial sales).

Why B is Correct:

Sales Transaction Type accurately represents new B2B purchases.

Enables separation of pricing logic by channel or buyer type.

Why Others Are Incorrect:

A: Quote Transaction Type is metadata used in CPQ quoting but not for pricing control logic.

C: Order Transaction Type applies post-sale; discounts must be calculated pre-order in pricing.

Salesforce CPQ Implementation Guide --- Volume-Based Pricing and Transaction Type Logic

Salesforce Subscription Management Implementation Guide --- Sales and Renewal Transaction Configuration

Salesforce Billing Implementation Guide --- Pricing Control via Transaction Type

Question 9

Question Type: MultipleChoice

A consultant is creating a decision table using a predefined template for product eligibility and availability. Which object types can the consultant use as evaluation criteria during product selection?

Options:

A- Product Price Book or Product Schedule
B- Product Relationship or Product Attribute
C- Product Qualification or Product Category Qualification

Answer:

C

Explanation:

Comprehensive and Detailed From Exact Extract:

In Salesforce Revenue Lifecycle Management, eligibility and availability rules are built using Decision Tables. The predefined templates for product eligibility and product availability rely on Qualification objects.

From the Revenue Lifecycle Management Implementation Guide:

''Eligibility and availability rules are driven by Product Qualifications and Product Category Qualifications, which determine whether a product can be selected for a specific customer or scenario.''

''Decision table templates for eligibility and availability use Qualification objects as the evaluation criteria.''

These objects are designed specifically for determining whether a product should be selectable based on business rules, customer attributes, and catalog categorizations.

Why other options are incorrect:

Product Price Book and Product Schedule are used for pricing, not eligibility.

Product Relationship and Product Attribute are used in configuration rules, not eligibility templates.

Salesforce Revenue Lifecycle Management Implementation Guide -- Eligibility and Availability Decision Tables; Product Qualification Framework.

==========

Question 10

Question Type: MultipleChoice

A product designer created the necessary products and bundles using Product Catalog Management. However, users are not able to see the products while preparing quotes.

Which action should the product designer take to resolve this?

Options:

A- Select the appropriate product selling model for each of the products.
B- Select the appropriate pricing procedure in the Salesforce Pricing Setup.
C- Select the appropriate context definition in the Product Discovery Settings.

Answer:

C

Explanation:

When using Product Catalog Management in Salesforce Revenue Cloud, visibility of products during quote creation (especially through Product Discovery) is controlled by the Context Definition. Contexts allow administrators to define when and for whom products are visible during the product selection process, based on criteria like sales channels, quote type, or other business rules.

If products are not appearing during quote preparation, it is often due to a missing or misconfigured context definition. Assigning the correct Context Definition in Product Discovery Settings ensures that products and bundles are available during quoting based on business logic.

Option A (selecting a selling model) affects how the product is priced and billed, but not its visibility.

Option B (pricing procedure) impacts price calculations, not product discovery or visibility.

Exact Extracts from Salesforce Revenue Cloud Documents:

CPQ Implementation Guide -- ''Product Discovery Configuration'':

''Ensure that context definitions are properly set so that products appear under the right conditions during quoting. Missing contexts will result in products not being visible to end users.''

Revenue Cloud Product Catalog Guide -- ''Using Context for Product Availability'':

''Context Definitions act as filters for product discovery. Without proper context mapping, products may not show up for selection.''

Salesforce CPQ Implementation Guide

Product Catalog Management Documentation

Revenue Cloud Product Discovery Settings Guide

==========

Question 11

Question Type: MultipleChoice

A product administrator needs to add a required rule using Constraint Modeling Language (CML) so that whenever a product called Desktop is added to a quote, another standalone product called Monitor will be automatically added.

What is the correct CML syntax to write this rule?

Options:

A- type Quote { relation desktop : Desktop[0..99]; relation monitor : Monitor[0..99]; constraint(desktop, monitor, 'Desktop requires Monitor'); }
B- type Quote { relation desktop : Desktop[0..99]; relation monitor : Monitor[0..99]; require(desktop[Desktop],monitor[Monitor], 'Desktop requires Monitor'); }
C- type Quote { relation desktop : Desktop; relation monitor : Monitor; require(Desktop[Desktop],Monitor[Monitor], 'Desktop requires Monitor'); }

Answer:

B

Explanation:

(150--250 words)

Constraint Modeling Language (CML) defines logical relationships between quote line items, allowing administrators to automate dependency and compatibility logic in Salesforce CPQ.

The keyword require() explicitly establishes a dependency that ensures one product must exist when another is present in a quote.

The correct syntax must define relationships with multiplicity ranges (e.g., [0..99]) and use the require() function, not constraint(), to specify the rule. Option B meets these criteria:

type Quote {

relation desktop : Desktop[0..99];

relation monitor : Monitor[0..99];

require(desktop[Desktop], monitor[Monitor], 'Desktop requires Monitor');

}

This ensures that when ''Desktop'' is added, ''Monitor'' is automatically included. The other options are incorrect because:

Option A uses the wrong function (constraint() instead of require()), which defines logical conditions but doesn't enforce automatic inclusion.

Option C omits multiplicity, which is required for valid relationship definition.

Exact Extract from Salesforce CPQ Implementation Guide:

''The require() statement in CML defines a dependency rule so that when one product is selected, the dependent product is automatically added to the quote.''

Salesforce CPQ Implementation Guide --- Constraint Rules and CML Syntax

Salesforce Revenue Cloud Developer Guide --- Defining Product Relationships in CML